08

III

E7.5-10326

CR-143063

1060B

THE UNIVERSITY OF KANSAS CENTER FOR RESEARCH, INC.
Space Technology Center—Nichols Hall
2291 Irving Hill Drive—Campus West
Lawrence, Kansas 66045

(E75-10326) KANSAS ENVIRONMENTAL AND RESOURCE STUDY: A GREAT PLAINS MODEL Final Report, 1 Aug. 1972 - 17 Mar. 1974 (Kansas Univ. Center for Research, Inc.) 32 p HC $3.75 CSCL 08F G3/43
N75-27514
Unclas
00326

KANSAS ENVIRONMENTAL AND RESOURCE STUDY: A GREAT PLAINS MODEL

Interpretation and Automatic Image Enhancement Facility

ERTS Interpretation and Automatic Image Enhancement Processing Support

R. M. Haralick, Principal Investigator
University of Kansas Center for Research, Inc.
Remote Sensing Laboratory
c/o Space Technology Center
Nichols Hall
2291 Irving Hill Drive-Campus West
Lawrence, Kansas 66045

February 12, 1975
Type III Final Report for the Period August 1, 1972- March 17, 1974

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND 20771
Contract NAS 5-21822, Task 2

A

KANSAS ENVIRONMENTAL AND
RESOURCE STUDY: A GREAT PLAINS
MODEL

FEBRUARY 1975

Type III Final Report for the Period
August 1, 1972 - March 17, 1975

Prepared for:

National Aeronautics and Space Administration
Goddard Spaceflight Center
Greenbelt, Maryland 20771

Contract No. NAS 5-21822

#5

Type III Final Report

for the period: August 1, 1972 - March 17, 1975

Title of Investigation: Interpretation and Automatic Image Enhancement Facility

ERTS A Proposal Number: 60-1

Task Number: 2

Investigator: R. M. Haralick

NASA-GSFC PI ID Number: UN 094

Report Prepared and Approved by:

Robert M. Haralick
Principal Investigator

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 2262-9	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Kansas Environmental and Resource Study: A Great Plains Model - Interpretation and Automatic Image Enhancement Facility - ERTS Interpretation & Automatic Image Enhancement Processing Support		5. TYPE OF REPORT & PERIOD COVERED Aug. 1, 1972-March 17, 1974 Type III Final Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert M. Haralick		8. CONTRACT OR GRANT NUMBER(s) NAS 5-21822, Task 2
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Kansas Center for Research, Inc. c/o Space Technology Center, 2291 Irving Hill Dr.-Campus West, Lawrence, Kansas 66045		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS National Aeronautics and Space Administration Goddard Space Flight Center Greenbelt, Maryland 20771		12. REPORT DATE February 12, 1975 13. NUMBER OF PAGES 24
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) National Aeronautics and Space Administration Goddard Space Flight Center Greenbelt, Maryland 20771		15. SECURITY CLASS. (of this report) 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This final report summarizes the kind of processing support provided to four Kansas ERTS investigations. Details of the processing results may be found in the respective final reports of these investigations.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

KANSAS ENVIRONMENTAL AND RESOURCE STUDY:
A GREAT PLAINS MODEL

ERTS INTERPRETATION & AUTOMATIC
IMAGE ENHANCEMENT PROCESSING SUPPORT

PREFACE

This final report summarizes the kind of processing support provided to four Kansas ERTS investigations. Details of the processing results may be found in the respective final reports of these investigations.

TABLE OF CONTENTS

LIST OF TABLES

ERTS INTERPRETATION AND AUTOMATIC IMAGE ENHANCEMENT PROCESSING SUPPOPT

KANDIDATS (Kansas Digital Image Data System) was used to support four of the Kansas ERTS investigations:

(1) Use of Feature Extraction Techniques for the Texture and Context Information in ERTS Imagery

(2) Wheat: Its Water Use, Production and Desease Detection and Prediction

(3) Extraction of Agricultural Wheat Statistics from ERTS

(4) Study of Monitoring Fresh Water Resources.

In each of the four investigations supported, the first processing task included accessing specified areas on the ERTS digital tapes, putting the data in the format desired by the investigators, and displaying the regions by a grey tone map created by overprinting characters on a digital printer. The ERTS texture study required the image data be formated in rectangular subimage blocks. The wheat production and disease detection study required a digital tabulation of the grey tones for each pixel on each MSS-band for many small areas on 18 different ERTS scenes listed in Table 1. The wheat identification study required histograms to be generated for many regions totalling over 8,000 square miles. The fresh water study required a grey tone spatial average to be taken on a few small areas over fresh water bodies for 13 different ERTS scenes listed in Table 2.

The fresh water investigation was the only one to make extensive use of the IDECS (Image Discrimination Enhancement Combination System) although all the investigations used it some. A thorough comparison between the results of the analog

processing done by IDECS and the digital processing done by KANDIDATS on the correlation between MSS grey tones and water quality showed the possibility of using either approach to obtain reasonable answers. However, it was clear that the digital approach was superior in accuracy. Spatial non-linearity of the analog IDECS system made repeatibility a slight function of scene orientation and context. If the analog approach is to be made more accurate, greater linearity of the video input devices are going to necessary.

For further information on the results obtained by the four investigations supported by automatic processing, a list of their final reports is presented.

Author	Report No.	Title	Period
R. M. Haralick	2262-9	Interpretation and Automatic Image Enhancement Facility	1 Aug. 1972 - 17 March 1974
H. L. Yarger	2265-10	Monitoring Fresh Water Resources	1 Aug. 1972 - 24 March 1974
E. T. Kanemasu	2263-9	Wheat: Its Water Use, Production and Disease Detection and Prediction	1 Aug. 1972 - 5 Jan. 1974
S. A. Morain	2264-9	Extraction of Agricultural Statistics from ERTS-A Data of Kansas	1 Aug. 1972 - 5 Jan. 1974

Image ID #	Coordinates* of Area Processed	Approx. Area (Sq. Mi.)	Image Date
1240-16523	(12, 42) (46, 66)	380	March 20, 1973
1022-16391	(98,112) (112,130)	117	August 14, 1972
1061-16564	(151,121) (279,249)	7632	Sept. 22, 1972
1058-16392	(98,112) (112,128)	104	Sept. 19, 1972
1061-16564	(125,117) (167,121)	98	Sept. 22, 1972
1021-16333	(87, 84) (108, 93)	88	Aug. 13, 1972
1076-16393	(52,102) (66,113)	72	Oct. 7, 1972
1291-16344	(85, 5) (99, 16)	72	May 10, 1973
1312-16520	(23, 50) (60, 72)	379	May 31, 1973
1312-16520	(33, 20) (47, 31)	72	May 31, 1973
1295-16573	(164,142) (178,156)	157	May 14, 1973
1240-16523	(14, 16) (32, 27)	92	March 20, 1973
1348-16514	(20, 19) (34, 30)	72	July 6, 1973
1348-16514	(13, 46) (50, 67)	362	July 6, 1973
1237-16345	(70, 2) (84, 13)	72	March 17, 1973
1274-16403	(66,128) (80,139)	72	April 23, 1973
1346-16395	(50,125) (64,136)	72	July 4, 1973
1256-16403	(70,140) (80,155)	70	April 5, 1973

* Given in Millimeters

Table 1 lists the ERTS scenes processed
for the wheat disease detection study.

Image ID #	Coordinates* of Area Processed	Approx. Area (Sq. Mi.)	Image Date
1002-16300	(17, 52) (29, 72)	110	July 25, 1972
1021-16333	(93, 103) (101, 115)	110	August 13, 1972
1022-16391	(100, 83) (115, 92)	160	August 14, 1972
	(110, 92) (120, 114)		
1039-16334	(83, 45) (95, 67)	110	August 31, 1972
1058-16392	(88, 16) (95, 28)	160	
	(98, 30) (108, 40)		Sept. 19, 1972
	(95, 14) (100, 32)	130	
1976-16393	(87, 17) (94, 32)	160	Oct. 7, 1972
	(92.5, 32) (108, 53)		
1094-16395	(92.5, 2) (100, 18)	160	Oct. 25, 1972
	(95, 18) (110, 29)		
1130-16401	(95, 5) (105, 21)	160	Nov. 30, 1972
	(100, 21) (115, 33)		
1256-16403	(112, 17) (122, 35)	160	April 5, 1973
	(118, 35) (131, 45)		
1274-16403	(110, 31) (122, 49)	160	April 23, 1973
	(118, 49) (131, 58)		
1291-16344	(101, 50) (115, 73)	110	May 10, 1973
1292-16402	(112, 25) (122, 40)	160	May 11, 1973
	(119, 40) (132, 51)		
	(96, 59) (111, 76)	130	
1328-16400	(112, 18) (124, 36)	160	June 16, 1973
	(119, 40) (132, 51)		
1346-16395	(111, 17) (122, 32)	160	July 4, 1973
	(117, 32) (131, 42)		
1382-16392	(110, 12) (117, 23)	160	August 9, 1973
	(112, 23) (123, 32)		
1399-16332	(100, 43) (113, 62)	110	August 26, 1973
1400-16300	(109, 18) (121, 36)	160	August 27, 1973
	(116, 36) (132, 46)		
	(93, 52) (107, 68)	130	

*Given in Millimeters

Table 2 lists the ERTS scenes processed for the water quality study.

APPENDIX I

IMAGE PROCESSING

Introduction

Image processing encompasses all the various operations which can be applied to photographic or image data. These include, but are not limited to, image compression, image restoration, image enhancement, preprocessing, quantization, spatial filtering and other image pattern recognition techniques. Interactive image processing refers to the use of an operator or analyst at a console with a means of assessing, preprocessing, feature extracting, classifying, identifying and displaying the original imagery or the processed imagery for his subjective evaluation and further interaction.

Image data are collected in the course of scientific experiments, medical tests, surveillance operations, and satellite and telescopic photography. To an increasing extent, these data must be processed by computer before they can be used by human interpreters. Computer processing has evolved over the last decade from the use of specialized machines to more flexible program packages using digital computers to sophisticated full scale computer systems. This move toward computer processing of images has been necessitated by the large volume of image data being produced at the present. One example of this production is the Earth Resources Technology Satellite which produces 782 images each day.

As a result, image processing software must resolve some of the most difficult problems confronting programmed systems: handling large amounts of data and heavy and varied computational loads. These problems require sophisticated and complex software for even the simplest image processing tasks.

Image Processing Systems

Image processing is the manipulation of image data. This manipulation generally includes all processes which remove degradation from, provide feature enhancement of, or extract information about the image data. At various points in this process it may be necessary to provide information about the progress made during the course of the processing. Despite wide variation in applications, the specific tasks that constitute image processing are common to many applications and can be found in most image processing systems. These basic functions form a set of utility operations which are necessary for any image processing system. This set includes image retrieval routines, image access routines and the various processing routines.

Image Creation

Once introduced to a computer, an image is ordinarily a two-dimensional sampling of intensities. This two-dimensionality reflects both the physical characteristics of most images and the rectangular nature of most scanning techniques. In all cases, the digitized version of the image represents only a portion of the information found in the original scene. This loss of information occurs first in the acquisition of the data and then in the digitization of this data. The spatial resolution and digital precision of the data grey levels specify to what extent information has been retained by the digital image. The digital image produced by the scanning process and A/D conversion, takes the form of a finite two-dimensional array of small regions called resolution cells and each resolution cell has an associated grey level.

Both the size of image data sets and the diversity of processing techniques present problems when organizing the format of these image data sets. Large, high-resolution image data sets are too large to be contained in main memory during processing and, therefore, they must be maintained instead in some manner on peripheral devices. By their nature, these devices require more time to access data than main memory. Once the decision has been made to store image data on some peripheral device, the question of data format arises. The format required is not necessarily obvious since the data structures themselves may restrict the types of image processing techniques which are easily implemented.

In addition, image data must ordinarily be compacted when placed on auxiliary storage devices to make maximum use of storage space. This means placing one or more digital values into one word to reduce storage requirements. This compaction makes the data unwieldy for use in computational purposes and therefore requires additional software to provide the ability to unpack the data and bring it into core in some more readily usable form.

Image Characteristics

Besides the actual grey levels associated with the digital image, there are various general attributes about the image that must be maintained. These include the area over which the imagery was taken, the sensor creating the image, various processing routines previously applied to the data along with their parameters as well as the more mundane parameters which allow the system to keep track of the image. These parameters include the image size, the number of images in the image set, the type of the data format, etc.

Traditionally, these attributes are passed through the system as parameters. The problems facing the operator with respect to keeping these parameters are not trivial. He must either know the parameters or where to find them. However, the nature of these attributes is such that they are fixed for any given image or set of images. Therefore, once they have been introduced to the system, they should be automatically maintained by the system. If this mechanism is an integral part of the system, significant information such as image size etc. may be easily remembered and transported between processing steps to provide necessary parameters and minimize repetitive user I/O. Additional information with regard to the image processing techniques and their corresponding parameters which have led to the current state of the image may be saved for later traces of image processing and comparison with other processes.

Software System Design

The software of an image processing system includes processing packages and a monitor section that directs the flow of control and information between the various subsystems. The processing of an image is a series of discrete steps, each yielding a product that may be used later in the sequence. The monitor initiates each step by calling it into action and providing the necessary information for it to run to completion.

The monitor is the point at which the operator and the image processing system are interfaced. The monitor and its subsystems should assume responsibility for as much bookkeeping as possible in order to free the operator from routine tasks. However it should be flexible enough to allow the operator to control these tasks if necessary. The monitor must also examine all operator input to insure to the largest extent possible that all his entries are free of error and inform him clearly of any that are not.

The amount of operator direction required by a system varies with the type of image processing being done. A system that is processing large numbers of images in a well-defined manner requires only minimal operator interaction. In contrast, when image processing is being done in a research environment where the goal is to determine the above mentioned "well-defined manner," the operator should be able to guide the program much more closely and readily. At this point, image processing enters the area known as interactive image processing. In this situation, the operator is the person who measures the results directly. He is in charge of the design and evaluation process and it is the function of the system to assist him as fully and as easily as possible with automation in an interactive/hands-on environment.

The goal of an interactive image processing system is to provide a quick flexible way of analyzing and manipulating image data and trying out various tools from a versatile tool kit to determine which algorithm should be selected for a given application. This evaluation of various tools is enhanced by interactive automation of known procedures so they can be applied in a routine fashion with immediate availability of the results to the operator.

The language by which the system is guided must be appropriate to the operator and the task. The operator should be required to provide only the minimum amount of information necessary to invoke his task and be given immediate feedback if his entry is in error. In addition, information should be available to prompt him on how to use the system. Of course, this information should be as concise, explanatory, and up to date as possible. Such an implementation should enable a new operator to receive necessary help but should not stand in the way of an experienced operator. The system must allow the operator to retrieve information directly, simply, and in an easily recognizable format.

The above mentioned properties force a similar construction on most image processing systems. The many programs necessary for implementing the various options

cannot all be stored in main memory at once. This generally leads to some form of overlay system where the monitor calls various subsystems into main memory as they are needed and overlays systems no longer necessary.

Description of Some Image Processing Systems

Several approaches to image processing systems have been developed by various organizations. Five of these systems are described in the next sections. These systems are software packages which are implemented on a particular configuration of a general purpose computer. The five systems discussed are LARSYS, IDAMS, ERIMS, VICAR, and IDIMS. A scientific programming language like FORTRAN is usually used for the major programs accessible to a user while a limited number of subroutines are normally written in assembly language to perform tasks that are done frequently and which cannot be efficiently programmed in a higher level language.

LARSYS Multispectral Data Processing System

LARSYS was developed by the Laboratory for Applications of Remote Sensing of Purdue University. It is a system of programs designed for the batch treatment of multispectral data and utilizes an IBM360/67 computer. All programs are controlled by a monitor, whose function is to recognize and interpret the monitor control cards, describing the required jobs, and to load the program supervisors.

Each program has its own supervisor, which recognizes and interprets the supervisor control cards, and coordinates and keeps under control the processors to perform the required analysis. The desired type of processing may be requested from a terminal by depressing the corresponding push-button combination on the keyboard.

LARSYS is capable of performing the following tasks:

(1) Inputting ground truth information;

(2) Calculate histograms, spectral plots, means, and correlation matrices;

(3) Determine the optimal set of spectral bands for classification of the available data and provide the initial appraisal for the separability of the classes involved;

(4) Implement a maximum likelihood classification scheme on a point-by-point basis and save this information for further evaluation;

(5) Display of image data;

(6) Spatial classification;

(7) Pictorial printouts of image data;

(8) Plot the spectral magnitude of data;

(9) Transf r specified classification files onto another tape;

(10) List identification information for files.

I AMS, Image Display and Manipulation System

IDAMS was developed by CSC, Computer Sciences Corporation, for NASA's Goddard Space Flight Center in August, 1972. It consists of a modular package of task routines which perform a wide range of image processing operations. IDAMS is designed to facilitate digitizing, displaying, enhancing, manipulating and recording pictorial information. These tasks can be combined in any way the user desires, as long as he provides for required inputs and outputs.

This system was designed to operate on the CDC-3200 computer. It allows a user to select an image, display it, zoom in on a selected portion, locate points of interest or select edge coordinates of a section for further processing. In addition, it contains a special purpose system checking package to aid design personnel in tracking hardware malfunctions to their sources.

IDAMS is capable of performing the following tasks:

(1) List an IDAMS tape on the line printer in integer format;

(2) Two-dimensional image convoluation;

(3) Enlarge an image using a set of weights to specify an arbitrary interpolation scheme;

(4) One or two-dimensional Fast Fourier transforms;

(5) Reduction of image size by straight averaging;

(6) Generation of histograms and image statistics;

(7) Printout of image data on line printer;

(8) Tape image editor;

(9) Window insertion and mosaic routine;

(10) IDAMS format to VICAR format;

(11) ERTS format to IDAMS format;

(12) Grid overlay generating program;

(13) Image correlation routine;

(14) Contrast stretch or modification.

The Computer Science Corporation developed a similar system called DIMES (Digital Image Manipulation and Enhancement Systems) for US Army Engineer Topographic Laboratories.

ERIMS Multispectral Data Processing Software

The Environmental Research Institute of Michigan developed a software package for extracting Earth Resources Information from Multispectral Scanner data. ERIMS programs are available on the IBM7094 computer.

The major characteristics of ERIMS include signal conditioning; a variety of classifiers such as maximum likelihood, best linear multiclass classifier, and adaptive classifiers; and post recognition processing including automatic area and perimeter measurements and classifier performance on known test sets.

ERIMS is capable of performing the following tasks:

(1) Produce a digital map on a line printer;

(2) Produce a printout of exact data values over a certain area;

(3) Generate histograms;

(4) Generate eigenvalues and eigenvectors of the covariance matrix;

(5) Examine a set of signatures to determine the best second best, etc. channels to be used for recognition classification;

(6) Determine how well separated a set of signatures are by calculating a pair-wise probability of misclassification between each possible pair of signatures;

(7) Combine the distributions of a set of signatures with optional weighting of the individual signatures or scaling of the signatures;

(8) Derive a correction for scan angle dependent variations in the data;

(9) Apply corrections to the data;

(10) Find the darkest object in each channel within a scene;

(11) Averages over all data points within rectangles of a specified size, separately in each channel, to produce an output having one data point for each such rectangle;

(12) Allows insertion of ground truth data to the data sets (Then this information can be used to automatically display the selected fields.);

(13) Miscellaneous supporting programs (These programs support the various mechanics of handling the data as opposed to performing any specified processing task.).

VICAR

The VICAR (Video Image Communication and Retrieval) system was developed by a cooperative effort of the Jet Propulsion Laboratory (JPL), Pasadena, California, and IBM in 1968. VICAR facilitates the acquisition, digital processing, and recording of image data. System objectives include ease of operation for scientific personnel not familiar with systems programming as well as the capability to perform digital image processing on a production basis.

VICAR is used to translate a relatively simple language to the more sophisticated needs of the operating systems of the IBM360-370 computers. The bulk of the VICAR system consists of packages of algorithmic programs that perform image processing tasks. These programs reside in a library and are invoked by the system in response to user directives. This library of programs can be extended by coding new algorithms in FORTRAN or assembly language. To be compatible with the VICAR system, VICAR input/output statements must be included with the normal FORTRAN input/output. The VICAR system maintains histories of processes and content-description labels on its permanent data sets.

VICAR is capable of performing the following image processing tasks;

(1) Find and replace resolution cells deviating by more than a specified tolerance from the average of corresponding pixels in adjacent lines;
(2) Compute statistics of up to 400 specified areas in up to 10 input images;
(3) Display images on a cathode-ray tube;
(4) Compare two images by forming a two-dimensional histogram of the intersection of image samples;
(5) Add or subtract two images;
(6) Fourier transformation;
(7) Expansion/Compression of image size;
(8) Overlay a graduated grid network on an image to locate pixel coordinates;
(9) Compute a grey-level frequency histogram of any portion of an image;
(10) Subtract a given noise pattern from an image;
(11) Compute a one-dimensional power spectrum;
(12) Generate an image;
(13) Average up to ten images;
(14) Produce a decimal printout of an image with a histogram;
(15) Combine two images of unequal size into one composite.

IDIMS

IDIMS was developed by the Electromagnetic Sensory Laboratory. It is an Interactive Digital Image Manipulation System designed to run on a Hewlett Packard 3000. Images can be referenced by name and a catalog containing information about the images (name, size, access frequency, location on disk or tape, etc.) is automatically maintained. The image processing functions are referred to by name and frequently used sequences of processing functions can be combined into a single new function with a simple MACRO command routine. All user interaction are done in the PECOS (Picture Enhancement Computer Operating System) language whose syntax is input image name, process, output image name. The system provides prompting for inputting regained processing function parameters. The multiband images are stored as single image files. The user can request any single band or sequence of bands when specifying the image name.

IDIMS can perform extensive mensuration functions which allow point selection and labeling, line and count fitting, line deletion, and length and area determination. It has a complete image graphics and display capability which allow grey level and pseudo-color mapping as a display function, graphics overlay on displays, contouring and image flickering.

IDIMS is capable of performing the following processing tasks:

(1) Image copying, expanding, and compressing;

(2) Left right, top bottom reversal, image transposition, and 180° image rotation;

(3) Image mosaicking, registering, congruencing, and combining;

(4) Image adding, dividing, multiplying;

(5) A variety of quantizing or intensity transformation;

(6) Image histograms;

(7) Fourier transforms, filtering, convolution, and frequency spectrum operations;

(8) Maximum likelihood statistic collection and classification;

(9) Clustering;

(10) Table look decision rules;

(11) Classification evaluation;

(12) Image display or printing;

(13) Divergence and Karhunen Loeve feature selecting.

KANDIDATS

Introduction

KANDIDATS (Kansas Digital Image Data System) refers to the entire image processing effort under development at the Remote Sensing Laboratory, University of Kansas. This has resulted in a standard digital image format along with several processing systems to manipulate this image data. These various processing systems include:

(1) Image data compression;

(2) Textural feature extraction;

(3) Bayesian classification;

(4) Spatial clustering algorithms;

(5) Image utility functions.

KANDIDATS Operation Overview

KANDIDATS is an interactive/batch-mode digital multi-image pattern recognition system designed to facilitate the man-machine interface between the user and the PDP-15/IDECS/IBM7094 II complex. It is intended to allow users with an interest in image processing and varied degrees of sophistication with computer hardware and software an easy access to the image processing facilities available. KANDIDATS provides a great deal of flexibility and degree of freedom to the operator. It is designed primarily as a tool to be used in a research environment. The type of image processing it provides is the flexible, relatively small quantity image processing tasks necessary in formulating and evaluating algorithms which will later be applied to large amounts of image data. KANDIDATS runs on the PDP-15/IBM computer system and uses the IDECS as an image acquisition and display device. Digital image data may be introduced through magnetic tape units on the IBM machine.

The entire system is guided by an operator at a console directing the system either by initiating commands or by directing input to be taken from a command file. KANDIDATS then manipulates the particular image data accordingly.

Once image data is entered into the system, KANDIDATS automatically maintains and processes multiple digital images in a standard format which provides a complete processing history for the image from the time it enters the system until the time it is no longer needed.

Image processing capabilities currently implemented in the KANDIDATS system include:

(1) Equal interval and equal probability quantizing;
(2) Sub-image cutting and pasting;
(3) Gradient operations;
(4) Spatial clustering;
(5) Computing textural feature images;
(6) Histogram and Scattergrum determination;
(7) Decision rule determination and image classification according to these decision rules;
(8) Image display;
(9) Image transformation;
(10) Image convolution;
(11) Expansion and Compression of image scales;
(12) Reformatting of images;
(13) Creation of images;
(14) Addition of ground truth information or maps to images.

There exists a set of specific image operations for each one of the abovementioned image processing tasks. The operator selects the appropriate operation by inputting commands to KANDIDATS via the teletype or the CRT terminal. The KANDIDATS package provides extensive error checking and frees the user from the bookwork and housekeeping necessary to set up these operations on the computer. The commands all have the same simple form and are decoded by the KANDIDATS command string interpreter. Each command string contains certain basic information:

(1) The abbreviated name of the operation;
(2) The name of the destination device on which to place the output image;
(3) The name of the created image;
(4) The name of the origination device on which to find the input image;
(5) The name of the input image.

The rationale for the command string being set up this way can be illustrated by an analogy with a busy office. In the office there are many workers each equipped with particular talents. The office has many kinds of file cabinets for storing information. The investigator gets work done by making a request of or by giving a command to one of the workers. He identifies for the worker which file cabinet and

the name of the folder in the cabinet where he can find the material which needs to be worked on. Then, he identifies for the worker which file cabinet and the name of the folder into which he must place the material he creates by working.

In KANDIDATS, every image has a name and is stored in a folder with that name. The cabinet into which the folder containing the image is placed is really the physical device on which the image resides. KANDIDATS has the following peripheral devices on which an image as well as other information can reside:

(1) IBM compatible magnetic tape drives;

(2) Disk pack (movable head);

(3) Card reader;

(4) IDECS disk.

The operator guides the various image data sets between the various devices applying various processing algorithms to them.

In addition, KANDIDATS allows the batch processing of image data through the creation of image processing task files. These task files known as KANDIDATS run files may be entered in the same format as that used when the operator enters command strings and responses at the teletype. This run file is stored on the disk and can be summoned at any point and used to instruct KANDIDATS in lieu of operator interaction.

KANDIDATS has a monitor which resides in memory throughout a KANDIDATS program run. This monitor takes user input in the form of a KANDIDATS command string, decodes it and calls in the appropriate subsystem to perform the designated function. The subsystem requests any additional information needed to perform the process and initiates the process. After the processing has begun, it is carried to completion unless the operator interrupts it or an error occurs. In all cases, return is made to the KANDIDATS monitor and appropriate action is taken. On normal return, KANDIDATS requests another command from the operator or run file. If the operator terminates the operation prematurely, return is made to the monitor and the termination noted. If termination occurs during processing from a KANDIDATS run file, the monitor asks for permission to continue or exist from the run file. When the system encounters a recoverable error, an error flag is set and the KANDIDATS monitor prints out a list of subroutine names whose calls led to the error along with the event number returned. The number can then be matched to an error list to determine the cause of the error. If the error occurs during run file operation, the operator is asked whether to continue with the operation or abort it.

KANDIDATS Command String Interpreter

KANDIDATS presently runs on the PDP-15 computer under the DOS monitor. KANDIDATS itself has a monitor which resides in memory throughout a KANDIDATS program run. This monitor controls the overlay of various programs at the users request. There is a set of specific image operations for each one of the above-mentioned image processing tasks. The user selects the appropriate operation by inputting commands to KANDIDATS via the teletype or CRT terminal. Each command has the same simple form and is decoded by the KANDIDATS command string interpreter. Each command string contains certain basic information in the following sequence:

(1) The abbreviated name of the operation;
(2) The destination device name;
(3) The name for the output image;
(4) The source device name;
(5) The name of the input image.

This results in the following general command format:

#:VERB DEST FILE1 EXT (FLAGS)_SOURC FILE2 EXT,FILE3 EXT (FLG)

where:

#:	are the KANDIDATS prompting characters, they are always printed when KANDIDATS is requesting command input.
VERB	is a valid KANDIDATS command (maximum of five characters). Use the -SVOC- command for the list of valid KANDIDATS commands.
DEST	is the destination device name. A destination device need not be given provided that only a verb or a verb and flags only are given. Otherwise, it is required.
FILE1 EXT	is the destination file name. At command level, this is always optional, however, it may be asked for later by certain command processors.
(FLAGS)	is a list of alphabetic characters in parentheses. The appearance of each letter in the list causes the corresponding logical flag (in an array of 26 such flags) to be set to true. The meaning of each flag is defined by the individual commands. The flag remains on until the next input request. Flags are always optional, but may appear immediately after the command name if nothing else is in the line, or immediately after the file name.

_	The underscore is the delimiter between the destination and the source halves of the command. This is required if and only if a source device name is to be given and a destination device name was given.
SOURC	is the source device name. This is required if and only if a '_' is given in the command. The name given must come from the same list as the destination device name.
FILE2 EXT	is the first source file name. Up to three source file names (separated by commas) may appear.
(FLG)	is the same as the first set of flags.

SYSTEM DEFINED FLAGS

These flags have the same meaning regardless of the funtion they are used with.

T - If in run-file mode, set the teletype input device to be the real teletype, otherwise, teletype input is taken from the run file.

L - Reset the teletype output file to be the IBM7094 line printer.

The command string is fashioned after the command string employed by DOSPIP, a DOS monitor utility program. This maintains continuity within the available user programs.

KANDIDATS command strings are limited to a maximum of 72 characters, a single console device input line, and no provision is made for command string continuation lines.

KANDIDATS commands are divided into three categories. These are described in the next three subsections.

Destination/Source Commands

Commands of this type are used to specify KANDIDATS operations which involve a transformation of image data resulting in a new system standard file. This generally involves an input and an output device and file name. Such commands consist of three sections: [VERB] [DESTINATION]_[SOURCE]. The elements within these sections specify where the data to be processed is to be found [SOURCE], the type of operation to be performed [VERB] and where the result of the operation is to go [DESTINATION].

The verb indicating the operation to be performed is always the first element in any KANDIDATS command string. For example, in the following command string:

#:EPROB DP LAWO32 IMG _ DP LAWERT IMG

the verb section consists of:

EPROB

The elements of this section specify:

(a) EPROB = an equal probability quantization operation is to be performed.

The destination section consists of:

DP LAWO32 IMG

(a) DP = the disk pack is to receive the quantized image information.

(b) LAWO32IMG = the SIF file name into which the quantized image data is to be written.

The source section of the above command string consists of:

DP LAWERT IMG

The elements of this section specify:

(a) DP = the image data to be quantized resides on the disk pack.

(b) LAWERT IMG = the SIF file name on which the input image data will be found.

Single-Device Commands

Commands of this type are used to specify KANDIDATS operations which involve only a single device and a file or set of files. Such a command must specify the operation to be performed and the device involved. Where needed, file names and option flags may also be added to the command string. For example, the command

#: RUN DP RUN FIL

specifies that a KANDIDATS run file is to be used for command string input.

Utility Subsystem Calls

Commands of this type are used to call in KANDIDATS subsystems which themselves have command strings or some type of control structure with a degree of operator interaction associated with them. For example, the command -#:EXSIF -- specifies that the utility subsystem EXSIF is to be called into memory. In this particular case, initial device default is to the teletype. Output to the IBM can be specified by =#:EXSIF IB- or issuing the appropriate command at EXSIF level. This similar form makes this type of command very similar to single-device commands. It is classified separately due to the kind of operator interaction required once the subsystem gains control.

Documentation of KANDIDATS Interacting Aids

KANDIDATS is designed to be able to instruct a user with minimum knowledge of the system. Once an operator has reached KANDIDATS level (indicated by '#:' and a bell) there exist four commands to instruct him on usage of the system. They are:

(1) MESG - This command provides a message indicating current system status. It also explains in detail the format of the command strings, the various device mnemonics and their logical unit numbers. File-naming considerations, run file techniques, and methods for gracefully interrupting processing are included as well.

(2) SVOC - This command provides a list of all the valid system commands.

(3) VOCA - Once the operator has discovered the command he wants or thinks he wants, he can use this command -VOCA- to receive a short 30 character description of the command.

(4) EXPL - This command provides all the information about the command as follows:

ACTION	Describes the primary function of the command.
DEST	tells the legal device types that may appear in the destination device field of the command and the use to be made of that device. If it is a disc file, it explains the type or format of the file that is produced.
SOURCE	tells the legal device types that may appear in the source device field of the command and the use to be made of that device. If it is a disc device, the number of file names required (maximum of three) is given along with the format or type of each file.
FLAGS	tells the alphabetic flags recognized by the particular command process subsystem and the action taken for each.
CTRLT	tells the action taken if the user types a 'CTRL T' on the keyboard during the execution of the command. The default action for all commands is to return to the '#:' level. If a run file is in operation, the operator is asked if he wishes to abort it or continue.

COMMENTS This section gives any remarks which might be helpful in using the command as well as references to external documentation on the command. If the command makes use of either the IBM7094 or the IDECS, detailed set-up instructions will be given here.

In addition to these four 'help' commands, the default mode of system interaction is the long mode. This form of interaction gives detailed explanations of all input needed.

Error Processing

Once the operator has entered a syntactically correct command string and the KANDIDATS monitor has checked that all input files exist, the appropriate processing subystem is called into main memory overlaying the command string interpreter and leaving only a skeleton KANDIDATS monitor in main memory. At this point, the subsystem notes the input already received by KANDIDATS monitor and requests from the operator any additional information needed to complete the task requested. Once the proper information is entered, the actual processing begins. This activity continues until it is either successfully completed or an illegal condition arises. At this point, return is made either to the KANDIDATS monitor or the utility subsystem monitor where appropriate error processing is done in a manner which indicates clearly to the operator any error conditions that occurred and any action taken by the system as a result of the error. The monitor then asks for the next command input line.

KANDIDATS List of Commands

Command	Description
ASMD	Add, subtract, multiply, divide images
AVSIF	Average image
BAYER	Determine Bayes rule
BAYES	Count, smooth, normalize, determine Bayes rule, and classify
BLK	Block format transform SIF
BRIEF	Request brief dialogue
BUILD	Create a command file
CHKBD	Generate checkerboard image
CLSFY	Classify
CLSTR	Cluster
CMBIN	Combine two SIF files
CNECT	Label maximum connected regions
CNTING	Contingency table
COUNT	Count second order marginals by category
ELOG	End line echo-back to command input
EPROB	Equal probability quantize SIF
ERRTE	Determine error rate and best bands
EXIT	Normal exit from run-file mode
EXPL	Print command explanations
EXSIF	Examine SIF file (Image Editor Function)
FILL	Fill regions
FIXR	Fast inverse transform

FLIP	Top-bottom flip on image
FREON	Normalize second order marginals by prior probabilities
FREOS	Smooth second order marginals
FXTR	Fast transform (FFT, FHT, SLANT, DLB, COSINE)
GDTI	Ground truth entry
GNIMG	Generate random images
GRAD	Make gradient image from SIF file
GTSDP	Make symbolic image from descriptor records
HRPLT	Category histograms and range plots
LOG	Echo following lines to output device
LONG	Request long dialogue
LPLCE	Compute Laplacian
LSTDR	List descriptor records of SIF file
MESG	Print message file
MHIST	Make histogram of SIF file
MKBAR	Create an N-tone SIF file
MRRROR	Left-right reverse on image
PAUSE	Suspend processing of run-file for operator interaction
PC	Principal component transform
QUANT	Equal interval quantization of SIF file
R180	Rotate image 180°
RCNV	Rectangular convolution of SIF file
RMS	Find RMS error of SIF file
RTATE	Rotate image
RUN	Initiate a command file

RUNIN	Take question input from run file
SBIMG	Subdivide SIF file
SPLIT	Split connected file
STOP	Immediate stop of KANDIDATS
SVOC	Short list KANDIDATS vocabulary
TERTS	Transfer ERTS image tape to SIF
TID	Transfer SIF file ID block
TLARS	Convert 'LARSYS' tape to SIF file
TRNSP	Transpose image
TRSLD	Threshold gradient image
TSIF	Transfer SIF file
TSKY	Convert SKYLAB tape to SIF file
TTIN	Take question input from TTY
TXTRE	Texture Transform
UFILL	Unfill regions
VOCA	Print full vocabulary
XPCMP	Expand/Compress SIF file